

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平8-101817

(43)公開日 平成8年(1996)4月16日

(51) Int.Cl. ⁶ G 0 6 F 15/16 9/44 9/46	識別記号 3 7 0 N 5 5 2 3 4 0 B	庁内整理番号 7737-5B 7737-5B	F I	技術表示箇所
--	-------------------------------------	------------------------------	-----	--------

審査請求 未請求 請求項の数5 O.L (全13頁)

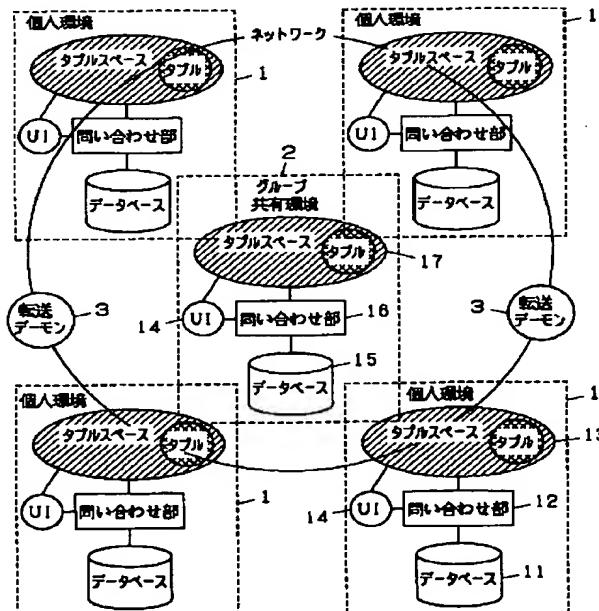
(21)出願番号 特願平6-237007	(71)出願人 富士ゼロックス株式会社 東京都港区赤坂三丁目3番5号
(22)出願日 平成6年(1994)9月30日	(72)発明者 林 祥一 神奈川県横浜市保土ヶ谷区神戸町134番地 横浜ビジネスパークイーストタワー 富士ゼロックス株式会社内
	(72)発明者 荒谷 徹 神奈川県横浜市保土ヶ谷区神戸町134番地 横浜ビジネスパークイーストタワー 富士ゼロックス株式会社内
	(74)代理人 弁理士 石井 康夫 (外1名)
	最終頁に続く

(54)【発明の名称】 ワークフロー支援システム

(57)【要約】

【目的】 自動的にタスクの実行順序を制御してワークフローを支援するワークフロー支援システムを提供する。

【構成】 ユーザインタフェース14から、個人または共有のデータベース11, 15中のワークフローのためのルールを選択し、その条件部をゴールとして問い合わせ部12に与えられると、問い合わせ部12はSLD反駁による計算を行なって、ワークフローが順次実行される。計算によりゴールの実行に失敗した場合には、失敗したゴールをタブルとしてタブルスペース17へ出力する。タブルはいずれかの問い合わせ部12, 16で解決され、結果がタブルスペース13に返される。タブルを実行するルールが存在しないとき、人間が処理を行ない、ユーザインタフェース14を介して処理結果を入力することにより、タスクを真としてワークフローの実行を継続する。



【特許請求の範囲】

【請求項1】 ネットワークによって接続された分散システムを用いてワークフローを支援するワークフロー支援システムにおいて、発生したタスクに対するルールの適用の可否を判定するための条件部とタスク列である実行部からなりあるタスクから次に実行すべきタスク列への展開を宣言的に記述したワークフローのためのルールを格納する複数のルール群格納手段と、該ルール群格納手段のそれぞれと対応して設けられ対応する前記ルール群格納手段に格納されたルールに含まれる前記タスクを起動するルール解釈実行手段と、起動された前記タスクに対して適合する条件部を持つルールをタスクの起動時に選択し選択されたルールを保持する前記ルール群格納手段と対応して設けられた前記ルール解釈実行手段に前記タスクを受け渡すルール選択手段を有し、前記ルール解釈実行手段は、前記ルール選択手段により受け渡されたタスクに適合する条件部を持つルールの実行部のタスクを順次起動することを特徴とするワークフロー支援システム。

【請求項2】 前記ルール解釈実行手段は、起動すべきタスクに対して適合する条件部を持つルールを選択する際に前記ルール群格納手段のいずれにも選択すべきルールが保有されていない場合には該タスクが起動できなかったこと第1の失敗状態として検出し、前記ルール解釈実行手段において第1の失敗状態が検出されたとき前記起動できなかったタスクの情報をユーザへ通知するとともに該通知に応じてユーザが行なう前記起動できなかったタスクの実行または実行の成果の入力を行なうユーザインターフェースを有していることを特徴とする請求項1に記載のワークフロー支援システム。

【請求項3】 前記ルール解釈実行手段は、起動したタスクが強制的あるいはシステムの定義した条件により第2の失敗状態となったことを検出し、第2の失敗状態となったタスクに対してすでに適用したルールとは別のルールであって適用可能なルールを再選択し、選択されるべきルールが存在しない場合にはこれを第1の失敗状態として検出し、該第1の失敗状態が回復しない場合には第1および第2の失敗状態となったタスクの起動順序における一つ前のタスクに後戻りし、該後戻りしたタスクを前記第2の失敗状態とすることを特徴とする請求項2に記載のワークフロー支援システム。

【請求項4】 前記ルールの条件部および実行部は該ルールの適用時およびタスクの起動時に情報の受け渡しを行なうための変数を用いることが可能であり、起動されたタスクと選択されたルールとの間で双方向に変数の値の受け渡しを行なうことができることを特徴とする請求項1に記載のワークフロー支援システム。

【請求項5】 前記ルール解釈実行手段は、ルールの実行部のタスク列中にある任意のタスクの実行が終了する前に、該実行中のルールのタスク列中のタスクを並行し

て起動することが可能であり、起動されたタスクに対して適合する条件部を持つルールを選択する際に値が未確定状態の変数の单一化の情報を一時保持しておき、任意のタスクで未確定状態の変数の値が確定された場合に、

05 前記单一化の情報に基づいて全てのタスクにおいて单一化された変数を同一値に確定することを特徴とする請求項4に記載のワークフロー支援システム。

【発明の詳細な説明】

【0001】

10 【産業上の利用分野】 本発明は、複数の情報処理装置がネットワークによって接続された分散環境におけるグループワークおよびワークフローの支援を行なうワークフロー支援システムに関するものである。

【0002】

15 【従来の技術】 一般にワークフローとは、仕事の流れを指す。例えば、会社の中で、ある人が物品の購入を行なう場合を考える。その人は、まず、購入伝票を作成し、上司のもとへ作成した購入伝票を持って行く。上司は購入伝票をチェックし、承認印を押して経理部門へ購入伝票を回す。経理部門では、購入伝票に基づき、物品を発注する。以降、まだまだ仕事は続くが、ここまで仕事の流れを見てもわかるように、購入伝票がある人からその上司へ、また、上司から経理部門へと流れ、仕事が進んで行く。このような仕事の流れをワークフローと呼んでいる。

20 【0003】 ここで、購入伝票を作成する仕事、購入伝票をチェックして承認印を押す仕事、物品を発注する仕事などは、それぞれが仕事の単位であり、各仕事の単位をタスクと呼ぶ。ワークフローは、タスクの列であるといいうことができる。また、各タスクは、さらに複数のタスクに分割されることもある。各タスクは、上述の例のように人が行なう仕事の場合もあるし、また、例えば、データベース検索のように、コンピュータが自動的に行なう仕事の場合もある。

25 【0004】 近年の情報処理装置の普及から、それぞれの仕事を行なう現場に端末などの情報処理装置を配置し、これらの複数の情報処理装置をネットワークで接続した分散環境が構成されている。各情報処理装置のユーザは、それぞれ受け持ちの仕事を行なうことになる。

30 【0005】 従来より、このような分散環境において、複数の情報処理装置のユーザにまたがるワークフローを支援するシステムの開発が進められている。それらのシステムは、多くの場合、電子メールに基づいている。電子メールとして送られるドキュメントの配達経路を定義することによってワークフローを定義する。例えば、上述の購入伝票の例では、購入伝票の作成者から上司を経て経理部門へ至る配達経路を定義し、購入伝票を電子メールとして送る。この電子メールに基づくシステムでは、基本的にドキュメントがユーザ環境を移動し、移動

45 した先で手続きを実行したり条件を評価することによつ

て、さらに次の行き先を一意に決定する。配送経路にいくつかの選択肢がある場合には、各々の経路に進む条件が定義され、ドキュメントの特定の部分などを状態として参照することによって、1つの経路を決定する。このような配送経路をグラフィカルユーザインタフェースを用いて定義するシステムもある。

【0006】このように、従来のシステムでは、ドキュメントの配送経路を自動的に決定することによって、ワークフローを実現するものであり、ドキュメント主導のシステムである。手続の管理については、全てサーバで管理するものや、各ユーザ環境ごとに管理するもの、あるいは配送されるドキュメントに埋め込むものなどがある。

【0007】しかしながら従来のワークフロー支援システムでは、電子メールが順に送られて行く宛先を最初から決定しておかなければならなかった。また、ワークフローのためにルールが手続的に記述することのできるシステムも開発されているが、やはり一意に宛先が決定するように、手続きの記述を行なわなければならない。このとき、例えば、次のタスクに進む前に値が確定しなければならない変数が各タスクごとに決定されている場合もあり、記述する手続きは複雑にならざるを得ないとともに、ワークフローのルールに記述されたタスクの実行順序に柔軟性を持たせて実行することができない。

【0008】また、あらかじめ想定し、ルールに記述しておくことができないような例外が発生した場合の対処が極めて困難であり、例外が発生した時点のみならず、以降のワークフロー全体に関してシステムの利用が不可能となるという問題もある。

【0009】

【発明が解決しようとする課題】本発明は、上述した事情に鑑みてなされたもので、あるワークフローの全体についてや、厳密にその条件や順序を知らなくても、ユーザが、各自の近傍の把握している条件や仕事のやり方をルール化するだけで、その部分が自動化でき、また、それらの部分フロー間や、まだルール化されていないために人間が行なったタスクとのつながりを管理し、自動的にフロー全体の実行や、成果物の受け渡しを制御しワークフローを支援するワークフロー支援システムを提供することを目的とするものである。

【0010】

【課題を解決するための手段】本発明は、請求項1に記載の発明においては、ネットワークによって接続された分散システムを用いてワークフローを支援するワークフロー支援システムにおいて、発生したタスクに対するルールの適用の可否を判定するための条件部とタスク列である実行部からなりあるタスクから次に実行すべきタスク列への展開を宣言的に記述したワークフローのためのルールを格納する複数のルール群格納手段と、該ルール群格納手段のそれぞれと対応して設けられ対応する前記

ルール群格納手段に格納されたルールに含まれる前記タスクを起動するルール解釈実行手段と、起動された前記タスクに対して適合する条件部を持つルールをタスクの起動時に選択し選択されたルールを保持する前記ルール

05 群格納手段と対応して設けられた前記ルール解釈実行手段に前記タスクを受け渡すルール選択手段を有し、前記ルール解釈実行手段は、前記ルール選択手段により受け渡されたタスクに適合する条件部を持つルールの実行部のタスクを順次起動することを特徴とするものである。

10 【0011】請求項2に記載の発明においては、請求項1に記載のワークフロー支援システムにおいて、前記ルール解釈実行手段は、起動すべきタスクに対して適合する条件部を持つルールを選択する際に前記ルール群格納手段のいずれにも選択すべきルールが保有されていない

15 場合には該タスクが起動できなかったことを第1の失敗状態として検出し、前記ルール解釈実行手段において第1の失敗状態が検出されたとき前記起動できなかったタスクの情報をユーザへ通知するとともに該通知に応じてユーザが行なう前記起動できなかったタスクの実行または

20 実行の成果の入力を行なうユーザインタフェースをしていることを特徴とするものである。

【0012】請求項3に記載の発明においては、請求項2に記載のワークフロー支援システムにおいて、前記ルール解釈実行手段は、起動したタスクが強制的にあるいはシステムの定義した条件により第2の失敗状態となつたことを検出し、第2の失敗状態となったタスクに対してすでに適用したルールとは別のルールであって適用可能なルールを再選択し、選択されるべきルールが存在しない場合にはこれを第1の失敗状態として検出し、該第

30 1の失敗状態が回復しない場合には第1および第2の失敗状態となったタスクの起動順序における一つ前のタスクに後戻りし、該後戻りしたタスクを前記第2の失敗状態とすることを特徴とするものである。

【0013】請求項4に記載の発明においては、請求項1に記載のワークフロー支援システムにおいて、前記ルールの条件部および実行部は該ルールの適用時およびタスクの起動時に情報の受け渡しを行なうための変数を用いることが可能であり、起動されたタスクと選択されたルールとの間で双方向に変数の値の受け渡しを行なうことができるることを特徴とするものである。

【0014】請求項5に記載の発明においては、請求項4に記載のワークフロー支援システムにおいて、前記ルール解釈実行手段は、ルールの実行部のタスク列中にある任意のタスクの実行が終了する前に、該実行中のルールのタスク列中のタスクを並行して起動することが可能であり、起動されたタスクに対して適合する条件部を持つルールを選択する際に値が未確定状態の変数の单一化の情報を一時保持しておき、任意のタスクで未確定状態の変数の値が確定された場合に、前記单一化の情報に基づいて全てのタスクにおいて单一化された変数を同一値

に確定することを特徴とするものである。

【0015】

【作用】請求項1に記載の発明によれば、あるタスクのワークフローのルールを、タスクの列に分割して、各々別のルールとしてルール群格納手段へ格納している。それぞれのルール群格納手段と対応して設けられたルール解釈実行手段では、対応するルール群格納手段に格納されているルールの実行部のタスクを順次起動する。タスクが起動されると、ルール選択手段は、そのタスクに対して適合する条件部を持つルールをタスクの起動時に選択する。選択されるルールは、タスクが起動されたルール解釈実行手段と対応するルール群格納手段内に存在することもあるし、他のルール群格納手段に存在することもある。ルール選択手段は、選択されたルールを有するルール群格納手段に対応したルール解釈実行手段に対して、起動されたタスクを受け渡す。ルール選択手段によってタスクを受け渡されたルール解釈実行手段は、選択されたルールの実行部のタスクを順次起動する。このようにして、最初のタスクを起動するだけで、宣言的に記述されたルールに基づいて、順次、タスクの起動およびタスクの受け渡しが行なわれ、複数のルール解釈実行手段が協調して動作し、ワークフローが実行されることになる。そのため、従来の電子メールのように、最初からすべての宛先を記述する必要はなく、ルール選択手段がタスクを実行することのできるルール解釈実行手段へ自動的にタスクを受け渡す。また、タスク実行時の条件分岐なども、従来のようにユーザが手続的に明示する必要はなく、いくつかのルールを宣言的に記述していざれかのルール群格納手段に格納しておくだけでよい。

【0016】また、本発明はネットワークによって接続された分散システム上に構築されているので、複数の端末装置を利用する複数のユーザあるいは複数のグループにまたがったワークフローの実行が可能となる。すなわち、すべてあるいは一部のルール群格納手段および対応するルール解釈実行手段の組を1人のユーザあるいは複数のユーザのグループと対応付け、ある時点のタスクの進行状態から、次のタスクを実行できるユーザあるいはグループを実行時に見つけ出し、見つけたユーザあるいはグループに対してタスクを依頼することができる。タスクを依頼されたユーザは、そのユーザの環境に定義されている仕事のやり方で、タスクを実行することができる。

【0017】請求項2に記載の発明によれば、起動されたタスクのためのルールがルール群格納手段のいずれにも保有されていない場合に、これを第1の失敗状態として検出し、その旨をユーザインターフェースを通じてユーザへ通知する。そして、ユーザによる処理を受けることによってタスクを完了させ、ワークフローを続行することができる。これにより、例えば、データベースを検索したり、アプリケーションを起動したりといったコンピ

ュータのみで実行するタスクの他に、人が行なう処理をも1つのタスクとして実行し、ワークフローを続行することが可能となる。

【0018】請求項3に記載の発明によれば、起動した

05 タスクが強制的に、あるいは、例えば、タイムアウトなどのシステムの定義した条件により第2の失敗状態となつたことを検出する。第2の失敗状態が検出された場合、そのタスクに対してすでに適用したルールとは別のルールであって適用可能なルールを再選択して実行する。選択されるべきルールが存在しない場合には、請求項2に記載の発明と同様にこのタスクを第1の失敗状態として検出し、ユーザによる処理を待つ。これによりタスクが完了すれば、ワークフローをそのまま続行することができる。第1の失敗状態が回復しない場合には、第15 1および第2の失敗状態となつたタスクの起動順序における一つ前のタスクにバックトラックし、バックトラックしたタスクを第2の失敗状態とする。すなわち、バックトラックしたタスクについて、別のルールを探索することになる。このバックトラックによって、ルール群格納手段に格納されているルールのうち適合可能なすべてのルールについて、その実行を試みるよう構成することができる。そのため、従来のように一意に宛先を指定する場合に比べ、柔軟なタスクの実行が可能である。また、従来の手続的な記述に比べ、複雑な実行順の制御でも簡単に記述することができる。

【0019】バックトラックの際には、一度にいくつのタスクまたは処理に後戻りすることも可能であり、後戻りが発生する位置と一度に行なわれる後戻りが終了する位置をルールの宣言時および実行中に指定しておけばよい。これにより、必要以上によけいな再試行をしないようにシステムを制御することが可能になる。

【0020】請求項4に記載の発明によれば、前記ルールの条件部および実行部には、変数を用いることが可能であり、ルールの適用時およびタスクの起動時に情報の受け渡しを行なうことができる。また、この情報の受け渡しは、起動されたタスクと選択されたルールとの間で双方向に行なうことができ、タスクの実行時に当該ワークフローのタスク間で、例えば、返信要求などの情報の交換の繰り返しを行なうことが可能となる。

【0021】この変数の値を未定のまま、タスクを起動することも可能である。このとき、起動されたタスクの側で未定状態の変数の値を確定することができる。例えば、請求項2に記載の発明のようにユーザがタスクを完了させる場合には、ユーザは、未完の処理や、自分が処理し、成果として入力しなければならない内容がどれであるかを知ることができ、それに対する処理を行なうことができる。

【0022】請求項5に記載の発明によれば、任意のタスクの実行が終了する前に、該実行中のルールのタスク列中のタスクを並行して起動することが可能である。こ

のとき、変数が未確定状態のままタスクが起動されることがある。この値が未確定状態の変数のうち、同じ変数について単一化の情報を一時保持しておき、任意のタスクで未確定状態の変数の値が確定された場合に、前記単一化の情報に基づいて全てのタスクにおいて単一化された変数を同一値に確定する。これにより、ワークフローのルールに記述されている手順にしたがって実行しているタスクにおいて、ユーザがそのタスクを終了していない中途の状態であっても、必要に応じてそれ以降のタスクを起動することができる。また、中途のタスクにおいて、作成されるべき各成果（文書）が完成するごとに、すでに起動されている後続のタスクへ前記成果を自動的に渡すことがルールを変更することなくできる。このように、ワークフローにおけるタスク列の進行方向と、その間の成果の受け渡しの方向とを一致させる必要がなくなり、同じワークフローのルールの中で、タスクの実行順序に柔軟性を持たせることができる。

【0023】この単一化された変数の確定は、上述の請求項3に記載の発明のようにバックトラックを行なう場合には、後戻りの対象となる各タスクにおいて、実行経歴上そのタスクで確定された変数のみを未定状態に戻すように構成すればよい。この過程を監視することによって、無効となる成果と、有効であり続ける成果とが、各自どれであるかをユーザは容易に知ることが可能となる。また、このようなタスクの実行とバックトラックの過程において、タスクの実行列と変数の確定値を記憶するとともに、後戻りによるタスクの冗長部を取り除くことによって、ユーザは、実行されたタスクあるいは処理の列と、そのワークフローに関係した文書、成果として有効である文書、関係した人などの情報を一覧することが可能になる。

【0024】

【実施例】図1は、本発明によるワークフロー支援システムの一実施例を示す構成図である。図中、1は個人環境、2は共有環境、3は転送デーモンプロセス、11、15はデータベース、12、16は問い合わせ部、13、17はタプルスペース、14はユーザインターフェースである。図1には、各構成を概念的に示している。物理的には、ネットワークに複数の端末装置が接続されている。ユーザは、各端末装置を操作する。個人環境1は、各ユーザに対して設定される環境であり、各端末装置に固有に設けられる必要はない。また、共有環境2も、ユーザのグループに対して設定される環境であり、特定の端末装置に設けられる必要はない。例えば、データベースは、ユーザが使用している端末装置に付属している記憶装置内に実体がある場合でも、実際にはネットワークを通じて他の端末装置に付属している記憶装置に実体が存在し、それらをアクセスする場合でもよい。

【0025】個人環境1には、その環境で作業を行なうユーザのための、個人用のデータベース11が存在す

る。このデータベースは、論理的に各個人ごとに管理されるものである。データベース11には、各自の個人用の情報や、ワークフローのルールなどが格納される。個人用の情報とワークフローのルールは、別のデータベースで管理することも可能であるが、以下の説明では、同一に管理されるものとして構成している。

【0026】グループ共有環境2には、複数のユーザを含むグループのための、共有のデータベース15が存在する。このグループは、運用上の都合で決められるものであり、システムがグループの決め方に制限を加えることはない。この共有のデータベース15には、グループで共有できる情報やワークフローのルールなどが格納される。このデータベース15も、論理的にグループで共有して管理されるものである。また、共有のデータベース15は複数存在して良く、各ユーザが複数のグループに含まれていても構わない。

【0027】ここで、ワークフローのルールとは、タスクのサブタスク列への展開を、宣言的に記述したものである。このワークフローのルールは、情報の受け渡しを行なう変数記述を含む条件部と実行部からなる。条件部が成り立つときに実行部に記述されたタスク列が先頭から順次起動される。実行部の各タスクは、起動されると実行待ちの状態となり、実行失敗あるいは実行中の状態を経て、実行終了時に達成あるいは拒否のいずれかの状態になる。タスクの実行は、別のルールの選択と実行、および、例えば、データベース11、15の検索、アプリケーションプログラムの起動などの処理や、ユーザインターフェース14を介した人間が行なう処理などが含まれる。

【0028】データベース11、15に格納される情報としては、各個人、各グループ用の情報や、ワークフローのルールの他に、各個人の属性に関する情報、組織に関する情報、ワークフローによって処理される文書などが格納される。

【0029】グループ共有空間2のデータベース15と、個人環境1のデータベース11に格納される内容は、特定のユーザのみが利用する個人用の情報と、グループ内の複数ユーザが共有したい情報とを分けて格納した方が望ましいというだけであり、運用上の都合によって使い分けられても構わない。すでに共有のデータベース15に存在するルールについて、個人的な理由によって、その一部を変更したルールなどを個人のデータベース11に保持することもできる。もちろん、一方のデータベースのみで構成することも可能である。

【0030】問い合わせ部12、16は、各データベースに対応して設けられ、ルールの起動や検索の指示を受

け、それらを解釈、実行する。例えば、グループ共有環境2のデータベース15から、承認者や、経理担当者を検索したり、購入伝票のフォームを検索するなどといった問い合わせを受け付け、これらの問い合わせに応えることができる。データベースの検索は、ワークフローのルールと他の情報とが同一のデータベースに格納されている場合には、後述するように、次に適用可能なワークフローのルールの検索方法と同様に、導出原理による推論によって求めることができる。ワークフローのルールと他の情報とを別のデータベースとして管理している場合には、ワークフローのルール以外の情報は、例えば、SQL, QBEなどの通常用いられているデータベースの問い合わせ言語処理系を用いることによって、効率よく検索を行なうことができる。また、問い合わせ部12, 16に、決まった動作をするルールをあらかじめ組み込んでおき、これによってアプリケーションを起動したり、各種の情報を検索するように構成することができる。

【0031】問い合わせ部12, 16としては、具体的には、SLD反駁による計算を行ない、論理型言語Prolog言語のインタプリタと同様な方法で実現することができる。Prologの言語仕様については、例えば、Clocksin, Mellish, Prologプログラミング、日本コンピュータ協会等を参照されたい。この場合、ワークフローのルールは、1階述語論理のホーン節の形式に分割して表現し、データベース1, 15に格納しておけばよい。

【0032】問い合わせ部12, 16は、ユーザインタフェース14からの問い合わせ内容、すなわち、指示されたワークフローのルールの条件部をゴールとし、そのゴールを証明するための導出論理による証明計算を開始する。問い合わせ部12, 16においては、ワークフローのルールの適用と、それによる実行部のタスク列は全て、最初に与えられたタスクを証明するための証明計算過程である。ゴールとなるタスクが達成状態になると、真であると証明されたことであり、拒否状態になると、偽であると証明されたことである。

【0033】また、ゴールとなるタスクの実行中や、以下に述べるタブルスペースに存在する間は、そのゴールの真偽値は未定状態である。そして、ゴールの真偽値を求めるこをゴールを解くといい、ゴールの真偽値が決められない（真とも偽とも証明できない）ときにゴールが解けない、あるいはゴールの実行に失敗するという。この証明の記録は、後戻り（バックトラック）などの冗長部を除外したワークフローの記録となり、変数の値は実際に当該ワークフローにおいて関与したユーザや文書となる。この照明の記録を保持することによって、ワークフローの一覧を得たり、ワークフローを追跡し、事務処理の効率化を行なうなど、種々の用途に用いることが可能である。

【0034】個人のデータベース11の中で、あるゴールが失敗した場合、すなわち、そのゴールにマッチする条件部を持つルールかデータがデータベース11内に存在しない場合には、問い合わせ部12は、後述するように、タブルスペースを通じて、この失敗したゴールの証明を他の問い合わせ部12, 16に依頼する。Prolog等の論理型言語においては、失敗したゴールは偽であることが証明されたと解釈するが、ここでは、その証明を他の環境あるいはユーザに依頼するものとする。

【0035】タブルスペース13, 17は、複数の問い合わせ部12, 16から読み書きができる共有空間である。グループ共有空間2のタブルスペース17には、問い合わせ部12において、個人用のデータベース11の中であるゴールが失敗した場合、その失敗したゴールがタブルとして書き込まれる。また、個人空間1のタブルスペース13には、例えばグループ共有空間2のタブルスペース17に出現したタブルが、転送デーモンによって転送されることにより書き込まれる。タブルスペース13, 17に書き込まれたタブルは、問い合わせ部12で読み出される。

【0036】転送デーモンプロセス3は、タブルスペース13, 17に保持されているタブルを他のタブルスペース13, 17に転送する。

【0037】ユーザインタフェース14は、ユーザからの仕事の依頼、および、タスクのうち人間が行なう処理の通知とその処理結果の入力等が行なわれる。ユーザが仕事を依頼、すなわち、ワークフローを起動するためには、ユーザインタフェース14から、例えば個人または共有のデータベース11, 15の中にあるワークフローのためのルールの条件部と同形のタスクを選択し、それをゴールとして個人環境1の問い合わせ部12に与えればよい。

【0038】図2は、本発明のワークフロー支援システムにおける個人環境1の問い合わせ部12の動作の一例を示すフローチャートである。問い合わせ部12は、マルチスレッド処理を行なうように構成することができる。すなわち、タスクの実行中であっても、タブルスペース13に出現したゴールは並列に実行される。

【0039】S21において、スレッドを二つ生成する。一方のスレッドは、ユーザから依頼されたワークフローを処理する。また、他方のスレッドは、タブルスペース13に出現したゴールを処理する。ユーザから依頼されたワークフローを処理するスレッドでは、S22において、ユーザからの問い合わせの入力を受け取る。そして、S23においてスレッドを1つ生成し、一方のスレッドではS24において、受け取った入力をゴールとしてSLD反駁による計算を行なう。

【0040】計算によりゴールの実行に失敗した場合には、S25でこれを検出し、S26で失敗したゴールをグループ共有環境2のタブルスペース17へ出力する。

そして、S 2 7において、失敗したゴールの結果が得られるまで、スレッドの動作を停止する。結果が返されると、その結果を読み取り、さらにS L D反駁による計算を続行する。また、S 2 3の他方のスレッドでは、別の問い合わせ入力に対応するため、S 2 2へ戻る。

【0 0 4 1】一方、タブルスペース1 3に出現したゴールを処理するスレッドは、S 2 8においてタブルスペース1 3にタブルが書き込まれるのを待つ。タブルがタブルスペース1 3に書き込まれると、そのタブルを読み取り、S 2 9においてタブルを処理するためのスレッドを1つ生成する。一方のスレッドでは、S 3 0においてS 2 4と同様に、読み取ったタブルをゴールとするS L D反駁による計算を行なう。このときゴールの実行に失敗した場合の処理についても、S 2 5ないしS 2 7と同様の処理をS 3 1ないしS 3 3において行なう。計算が終了すると、タスクの依頼元へ結果を返し、処理を終了する。また、他方のスレッドでは、別のタブルがタブルスペース1 3に書き込まれた場合に対処するため、S 2 8へ戻り、停止状態となる。

【0 0 4 2】このようにして、各個人環境1の問い合わせ部1 2では、ユーザから依頼された複数のワークフローの処理と、タブルスペース1 3に出現したゴールを並行して処理することができる。

【0 0 4 3】グループ共有環境2の問い合わせ部1 6の動作のうち、他のグループ共有環境2の問い合わせ部1 6や個人環境1の問い合わせ部1 2からタブルスペース1 7に書き込まれたタブルに対する処理、および、ある場合にはグループ共有環境2のユーザインタフェース1 4から受けた問い合わせに対する処理は、図2に示した個人環境1の問い合わせ部1 2の処理と同様である。ただし、グループ共有環境2の問い合わせ部1 6から出力されるタブルは、タブルスペース1 7に書き込まれる。しかし、タブルスペース1 7に書き込まれたタブルは、そのタブルを出力した問い合わせ部1 6では処理されない。かわりに、以下の2つの方法のどちらかによって処理される。

【0 0 4 4】まず1つめの方法は、そのゴールを解くことができるユーザが、ユーザインタフェース1 4を通じて自分でそのゴールを取り出し、自分のゴールとして処理を行ない、結果をユーザインタフェース1 4から入力するという方法である。グループ共有環境2のタブルスペース1 7に存在するタブルは、基本的にはこのグループのユーザがユーザインタフェース1 4を通して、参照

タスク（?人1, 物品購入処理）←

タスク（?人1, 購入伝票作成, ?購入伝票),
 タスク（?人1, 承認処理依頼, ?購入伝票, ?承認印),
 タスク（?人1, 発注処理依頼, ?購入伝票, ?承認印).
 タスク（?人1, 承認処理依頼, ?購入伝票, ?承認印)←
 承認者（?人1, ?人2),
 タスク（?人2, 承認処理, ?購入伝票, ?承認印),

することができる。このとき、余計なものがたくさん見えると不便であるので、ユーザインタフェース1 4にフィルタの機能を持たせることができる。各ユーザがフィルタの条件を設定しておくことにより、各ユーザに関係するものだけが見えるようにこうせいすることができる。あるいは、個人環境1の問い合わせ部1 2がグループ共有環境2のタブルスペース1 7を常に参照し、解決可能なゴールを取り出して実行するように構成することもできる。

- 05 10 15 20 25 30 35 40
- 【0 0 4 5】もう1つの方法は、転送デーモンプロセス3を用いて、タブルを他のタブルスペース1 3, 1 7に移動させる方法である。この方法は、データベース1 5内のデータから、そのゴールを解くことができるユーザか、解くことができるユーザが存在する他のグループを一意に特定できた場合や、そのデータベースを参照できるユーザが誰も解くことができない場合、あるいは解くことができるユーザが複数存在するにもかかわらず、一定時間経っても誰もそれを実行しようしない場合などに行なわれる。
- 【0 0 4 6】この場合は、転送デーモンプロセス3は、タブルを実行すべきユーザを調べ、そのユーザのタブルスペース1 3, 1 7にタブルを移動させる。あるいは、データベース1 5内の情報から、実行が可能であるユーザが複数存在することが判明し、なおかつ一定時間経つても誰もそれを実行しようしない場合は、実行可能なユーザの誰かのタブルスペースにそのゴールを移動させ、実行を促す。
- 【0 0 4 7】以下、具体例を用いて上述の動作の一例を説明する。ここでは、ある人が物品の購入処理を行なう場合の例を取り上げる。物品を購入する際には、購入伝票を作成し、承認者の承認印を受けた後、経理部門が発注するものとする。すなわち、「ある人1が物品購入を行なう場合には、購入伝票を作成し、承認処理を承認者に依頼し、経理担当に発注処理してもらう。」ことをワークフローのルールとして定義しておけばよい。これをワークフローのルールとしてグループ共有環境2のデータベース1 5に格納しておけばよい。
- 【0 0 4 8】ここでは、次の4つのワークフローのルールが宣言的に記述され、グループ共有環境2のデータベース1 5に格納されているものとする。これらのワークフローのルールは、例えば、ユーザインタフェース1 4を用いて定義することができる。

タスク（？人1, 発注処理依頼, ?購入伝票, ?承認印）
 タスク（？人1, 発注処理依頼, ?購入伝票, ?承認印）←
 経理担当（？人2）,
 タスク（？人2, 発注処理, ?購入伝票, ?承認印）
 タスク（？人1, 購入伝票作成, ?購入伝票）←
 フォーム取り出し（購入伝票作成, ?フォーム）,
 タスク（？人1, 記載, ?フォーム, ?購入伝票）.

【0049】上述の各ルールにおいて、「←」の左辺が条件部、右辺が実行部である。各ルール中の「？」が先頭についている文字列は変数である。なお、ルールの表記上、変数のスコープは、その変数が出現する1つのルール内のみである。そのため、上述の4つに分割されたルールは、各々独立に格納し、管理することができる。

【0050】例えば、上述の購入伝票処理をH氏が起動すると、H氏の個人用のデータベース11に対して、「タスク（H, 物品購入処理）」をゴールとする問い合わせが発生する。

【0051】H氏の個人環境1の問い合わせ部12は、変数の確定に関して、双方向性を持つパターンマッチングによって、ゴールにマッチする条件部を持つルールを検索し、そのルールを適用し、実行部を新たなゴールと

タスク（？人1, 物品購入処理）←
 タスク（？人1, 購入伝票作成, ?購入伝票),
 タスク（？人1, 承認処理依頼, ?購入伝票, ?承認印),
 タスク（？人1, 発注処理依頼, ?購入伝票, ?承認印）.

のルールがマッチする。すなわち、変数「？人1」の値は「H」となり、条件部「タスク（？人1, 物品購入処理）」と「タスク（H, 物品購入処理）」がマッチする。ルールの検索は、登録されている順序に従って行なわれる。

【0054】取り出されたルールが適用され、実行部の各タスクが順に実行される。まず、「タスク（H, 購入伝票作成, ?購入伝票）」を実行する。ここでさらに、このゴールにマッチするルールを検索し、適用する。ここでは、

タスク（？人1, 購入伝票作成, ?購入伝票）←
 フォーム取り出し（購入伝票作成, ?フォーム）,
 タスク（？人1, 記載, ?フォーム, ?購入伝票）.
 のルールとマッチし、このルールが適用される。このとき、変数「？人1」の値は「H」となる。このルールの実行部の各タスクが順次実行される。

【0055】まず、「フォーム取り出し（購入伝票作成, ?フォーム）」が実行される。このタスクは、文書データベースから購入伝票フォームの抽出を行なう処理を示している。システムは、文書データベースから購入伝票フォームを取り出し、この購入伝票フォームを変数「？フォーム」の値として、「フォーム取り出し（購入

タスク（？人1, 記載, ?文書フォーム, ?文書実体）←
 コピー（？文書フォーム, ?文書実体）,
 ワープロ起動（？文書実体）.

して展開しようとする。これによって、実行時に適用されるルールが定まり、ワークフローが実現される。

- 10 【0052】ここで、H氏のデータベースに物品購入処理のルールが存在しないものとすると、ゴール「タスク（H, 物品購入処理）」は失敗する。この場合、このゴールはグループ共有環境2のタプルスペース17にタプルとして出力される。
- 15 【0053】グループ共有環境2の問い合わせ部16は、タプルスペース17に書き込まれたタプルを取り出し、データベース15内のルールを検索する。グループ共有環境2のデータベース15には、上述した物品購入処理の4つのルールが格納されている。そのため、その20 ルールのうちからマッチするルールを取り出す。この場合には、

- 25 タスク（？人1, 購入伝票, ?購入伝票),
 タスク（？人1, 承認処理依頼, ?購入伝票, ?承認印),
 タスク（？人1, 発注処理依頼, ?購入伝票, ?承認印）.
 伝票作成、購入伝票フォーム）」が返される。
- 30 【0056】次に、「タスク（？人1, 記載, ?フォーム, ?購入伝票）」を実行しようとする。この時、変数「？人1」の値は「H」、「？フォーム」の値は「購入伝票フォーム」であるので、「タスク（H, 記載, 購入伝票フォーム, ?購入伝票）」を実行しようとする。しかし、このタスクについてのデータはグループ共有環境2のデータベース15には存在しない。上述のように、失敗したゴールはタプルとしてグループ共有環境2のタ

- 35 プルスペース17に出力される。そのため、「タスク（H, 記載, 購入伝票フォーム, ?購入伝票）」がタプルスペース17に出力される。この例の場合、タスクの実行者となるユーザは第1引数の「H」氏であることがシステムにより認識され、このタプルはH氏の個人用のタプルスペース13に移される。この移動は、グループ共有環境2の問い合わせ部16、あるいは、転送デーモンプロセス3によって行なわれる。
- 40 【0057】タプルがH氏のタプルスペース13に転送されると、問い合わせ部12がこれを検知して、タプル45 を読み取り、実行しようとする。H氏の個人用のデータベース11には、次のようなルールが存在するものとする。

- 45 【0057】タプルがH氏のタプルスペース13に転送されると、問い合わせ部12がこれを検知して、タプルを読み取り、実行しようとする。H氏の個人用のデータベース11には、次のようなルールが存在するものとする。

上述の「タスク（H, 記載, 購入伝票フォーム, ?購入伝票）」は、このルールとマッチする。このとき、変数「?人1」の値は「H」、「?文書フォーム」の値は「購入伝票フォーム」である。また、変数「?文書実体」は「?購入伝票」と対応づけられる。「タスク（H, 記載, 購入伝票フォーム, ?文書実体）」のルールが適用され、その実行部が順に実行される。まず、「コピー（購入伝票フォーム, ?文書実体）」が実行され、続いて「ワープロ起動（?文書実体）」が実行される。これらの実行は問い合わせ部12によって、システムのコマンドやアプリケーションの起動に置き換えられて実行される。

【0058】一方、このようなルールをH氏が用意しておらず、適合するルールが存在しない場合には、このゴールはH氏の個人環境1の問い合わせ部12において必ず失敗する。しかし、このゴールは、転送デーモンプロセス3によってH氏の個人環境1に要求されたタスクであるので、再びグループ供給環境2のタブルースペース17に戻されることはなく、代わりにユーザインタフェース14を通じてこの失敗したゴールをユーザであるH氏に通知する。このとき、ユーザインタフェース14は、変数になっている部分、すなわち、「?購入伝票」の部分を作成されるべき文書としてユーザに通知する。この例の場合は、H氏はユーザインタフェース14を通じて、まず、購入伝票の文書実体を作成し、記載しなければならないことを知ることができる。そして、購入伝票の文書実体を作成する作業をH氏自身が行ない、その成果物である購入伝票がどの文書であるかをユーザインタフェース14を通じて入力し、ルールが存在した場合と同様の処理に戻ることができる。購入伝票の文書実体は、例えば、ワープロで購入伝票フォームに所定事項を記載すればよい。

【0059】H氏が起動したワープロで「購入伝票文書」の記載を終了し、その文書をユーザインタフェース14から入力すると、問い合わせ部12は、「タスク（H, 記載, 購入伝票フォーム, ?文書実体）」は証明されたと解釈し、真であると結論する。そして、このタスクの証明の終了とともに、もともとのゴール「タスク（H, 物品購入処理）」を直接展開した最初のゴール「タスク（H, 購入伝票作成, ?購入伝票）」が証明されたことにもなり、これも真となる。さらに変数の確定値として、H氏が記載した「購入伝票文書」が変数「?購入伝票」の値となる。

【0060】また、変数は再び変数を部分的に含む値であっても、それを値として確定できる。例えば、変数「?購入伝票」の値を確定する際に、ユーザインタフェース14を通じて「購入伝票文書」ではなく「購入伝票文書、返信要求（?返信文書）」などを値とすることにより、「購入伝票文書」を以降のタスクに送るだけではなく、「返信要求（?返信文書）」により受け取ったこ

との返信を要求することもできる。このとき、ルールを変更する必要はない。そして、確定した変数の値は、その変数を持つタスク全てに伝わるので、その返信を「?返信文書」に受け取ることができる。

05 【0061】続いて、もともとのゴール「タスク（H, 物品購入処理）」を直接展開した次の「タスク（H, 承認処理依頼, 購入伝票文書, ?承認印）」、および、「タスク（H, 発注処理依頼, 購入伝票文書, ?承認印）」の処理を行なう。この場合にも、グループ共有環境10 2のデータベース15を検索して適合するルールを実行する。適合するルールがない場合には、上述のように、グループ共有環境2のタブルースペース17にタブルが出力され、H氏の個人用のデータベース11や他の共有あるいは個人用のデータベースを検索する。検索できなか15った場合は、H氏あるいはタスクを解決可能なユーザに對してタスクの解決を依頼することになる。また、ルール中の例えば「承認者（H, ?人2）」などのデータベース検索についても、やはり個人データベースで検索できなかった場合は、同様にグループ共有環境2のタブル20スペース17に出力され、共有のデータベース15上で同様の検索が行なわれる。このような動作が繰り返し行なわれることにより、ワークフローが実行される。

【0062】ここで、例えば、この物品購入処理を行なう際に、急いでいるにもかかわらず、承認者が出張中で25不在であった場合を考える。このような場合には、タスクの記述されている順序に従って実行されるので、承認処理でストップしてしまい、発注処理が遅れてしまうことがある。本発明のワークフロー支援システムでは、実行順序に関して柔軟性を持たせている。以下、この実行30順序の柔軟性の実現手段について説明する。ここでは、取りあえず承認印のないまま経理部門まで購入伝票を先に送り、発注処理に必要な情報収集や伝票の作成などをあらかじめ処理しておいて欲しいとする。

【0063】この場合は、ユーザインタフェース14を通じて、強制的に次のタスクへ進めることができる。これは、問い合わせ部12に対して、仮に「タスク（?人1, 承認処理依頼, ?購入伝票, ?承認印）」が証明されたという入力を与えればよい。問い合わせ部12は、このゴールが証明されたので、以降の計算を実行し、購入伝票を経理部門に回す。すなわち、タスク列の次のタスクを実行する。

【0064】後に承認者が出張から戻り、ユーザインタフェース14を通じてこのタスクの承認処理を拒否したい場合には、承認処理の状態を拒否状態とすれば良い。

45 タスクが拒否状態となると、このタスクのゴールが偽であると証明されたと解釈し、以降の証明を無効として、ここより後に確定された変数を未定状態に戻し、後戻りする。

【0065】一方、後から承認者が購入伝票を承認した50ときには、変数「?承認印」の値が「承認印」に確定さ

れる。本発明においては、ルールの変数は全て論理変数として扱われ、変数どうしの単一化も管理している。そのため、変数「?承認印」の値が「承認印」に確定されることによって、すでに実行中の発注処理のタスクの中で変数「?承認印」と単一化されている全ての変数の値も「承認印」に確定される。したがって、このようにタスクの起動と、成果物のシステムへの入力順序が入れ替わっても、ルールを変更する必要がなく、同じルールを用いることができる。また、このようにルールの実行順序を入れ替えて実行させることが可能である。

【0066】上述の具体例において、先の場合と同様に、この物品購入処理を行なう際に、急いでいるにもかかわらず、承認者が出張中で不在であった場合、今度は、別の承認者に代わって承認処理を行なってもらう場合も考えられる。このような場合には、承認処理のタスクを偽として例外処理を実行させることにより実現することができる。ここでいう例外処理は、単に別の選択肢を選び直す処理にしか過ぎない。この意味で、通常のシステムにおいて、システムに記述してある条件分岐による選択肢の中から、通常は選ばれない選択肢が選ばれた

タスク（?人1, 承認処理依頼,
承認者（?人1, ?人2）,
|,

タスク（?人2, 承認処理,

ここで、' | ' はコミットオペレータと呼ぶものであるが、その働きは Prolog 言語におけるカットオペレータに等価である。このオペレータの右側まで実行が進んだ後、何処かでゴールが拒否状態となり、後戻りによって再びこのオペレーターの位置へ戻ってきた場合には、このルールが適用される原因となったタスクの拒否状態が確定し、このタスクをゴールとして呼び出した 1 つ前のゴールの再試行に後戻りする。すなわち、上述の例では、最初の承認者が承認しないといえば、他の承認者を探すことなく、その決定が確定される。そして、

「タスク（?人1, 承認処理依頼, ?購入伝票, ?承認印）」の別の可能性が試行されることになる。他に適合するルールが存在しない場合には、「タスク（?人1, 購入伝票作成, ?購入伝票）」の別の可能性が試行される。この際に、変数「?購入伝票」の値の確定は最初のこのゴールの実行前に戻されるので、未定状態になる。すなわち H 氏は購入伝票の書き直しを要求されることになる。

【0070】ただし、論理型言語などの場合とは異なり、本発明による問い合わせ部 1・2, 1・6においては、通常、このコミットオペレータはルールには記述しなくてもよい。ルールはその同じルールによって起動される全ての実行例のひな型となるものである。そのため、ルール中にコミットオペレータを記述することは、全ての実行例に対して、規定してしまうので、コミットオペレータをルールに記述しなくてもよいように構成してい

場合の処理のことを例外処理と呼ぶのとは相違する。

【0067】通常の承認者が不在で承認が受けられずに処理が滞っている場合、承認処理のルールの実行部に記述されているタスクのうち、「タスク（?人2, 承認処理, ?購入伝票, ?承認印）」が実行中となっている。この状態において、ユーザインタフェース 1・4 を通じて、この「タスク（?人2, 承認処理, ?購入伝票, ?承認印）」が偽であると証明されたという入力を与える。あるいは一定時間たってもこのタスクが受理されず、真であるとも偽であるとも証明されない場合はシステムが自動的に偽であると証明する。

【0068】問い合わせ部 1・2 は、論理型言語と同様の手法によりバックトラックし、タスク列の 1 つ前の「承認者（?人1, ?人2）」の別の可能性を検索する。すなわち、後戻りして別の承認者の検索をやり直し、新たな承認者に依頼する。

【0069】このような後戻りが発生するとき、本発明のワークフロー支援システムでは、後戻りする位置を指定することができる。例えば、上述した承認処理の依頼 20 のルールを次のようにも記述することができる。

タスク（?人1, 承認処理依頼, ?購入伝票, ?承認印） ←

タスク（?人2, 承認処理,

?購入伝票, ?承認印） .

25 る。

【0071】コミットオペレータは、ルールに記述されていなくても、ユーザが実行時あるいは実行中にユーザインタフェース 1・4 を通じて、起動した具体的なワークフローごとに必要に応じて動的に挿入することができる。上述の例では、ルールにコミットオペレータがないほうが望ましく、最初の承認者がいないために一定時間経っても処理されない場合には、1 つ後戻りして他の承認者に転送できるようにルールを定義しておくことが望ましい。承認者がそのタスクを処理し、関与したときはコミットオペレーターが挿入され、そして承認を与えない場合は、承認処理そのものが拒否状態となるようにすればよい。

【0072】このように、本発明のワークフロー支援システムによれば、仕事の進み具合のある同じ状態から、40 以降のワークフローが数通り考えられる場合に、それら全てのルールをデータベース 1・1, 1・5 に格納しておくことができる。そして、選択したワークフローによるタスクが拒否状態になった場合にはバックトラック機構による後戻りによって、別の選択肢を選び直すことができる。したがって、あらかじめ想定できる例外についてのルールは、通常のルールと全く同等な形式で定義することができる。すなわち、この意味での例外は、本発明によるシステムにおいて、単に別の選択肢を選び直す処理にしかすぎない。

50 【0073】もちろん、システムに記述していない事態

が発生した場合の例外も発生する。ここで、システムに記述していないということは、処理が記述されていないだけでなく、ある条件の場合の選択肢そのものの記述がないことをいう。従来のシステムでは、このような状態はあってはならない状態であり、システムはエラーを起こして停止する。従来のシステムでは、このような意味での例外は、システム全体を閉じた系として実現しているために、記述されていない状態の発生は異常事態となる。本発明によるシステムにとっては、開放系が前提となっているので、これらは単に別の環境へ処理を依頼すれば済む処理であるに過ぎない。

【0074】例えば、通常行なわれているワークフローのあるゴールが拒否状態となり、それに代わる選択肢も存在しない場合、すなわちシステム全体に格納されているデータやルール以外の処理が必要となる場合、例外が発生する。このような例外が発生した場合には、その例外が発生したタスクはユーザである人間が対処しなければならないタスクである。本発明のワークフロー支援システムでは、すでに上述したように、このような例外発生の場合には、人間の対処を取り込み、以降のワークフローを継続することができる。

【0075】上述したように、本発明における問い合わせ部12、16は、ここでいう例外処理であるか否かにかかわらず、通常の処理においても、対応するデータベース11、15の検索で失敗したゴールは、グループ共有環境2のタブルスペース17に出力し、そのゴールの証明を他の環境に依頼している。人間が対処しなければならなかった例外処理は、どのデータベース11、15でも失敗し、タブルスペース17に残ってしまったゴールである。そして、人間が行なった対処は、そのゴールを真であると証明するための証明過程であったことになる。

【0076】通常の処理においても、ゴールをタブルスペース17へ出力した側の問い合わせ部12は、そのゴールの真偽の結果が判明するのを待っているだけで、どの環境でどのように証明されたかについては、知る必要がない。すなわち、人間にとての例外に対する処理は、システムにとっては全く例外ではなく、そのゴールを解いた他の環境のデータベースとその問い合わせ部がたまたまユーザである人間であったというだけのことには過ぎない。

【0077】このように、ルールが記述されていないことによって発生する例外に対しても、他のデータベース

や、ユーザである人間の処理をも含めることによって、柔軟に対応することができる。また、これによって、人間の行なう処理をも含めたワークフローの管理および支援を可能としている。

05 【0078】

【発明の効果】以上の説明から明らかなように、本発明によれば、ユーザは最初のタスクを起動するだけで、宣言的に記述されたルールに基づいて、順次、タスクの起動およびタスクの分割が行なわれ、ワークフローを実行、管理することができる。実行時に、分割されたワークフローのルールの中から次に適用可能なルールを検索し、適用することにより、従来の電子メールのように、最初からすべての宛先を記述したり、条件判断による分岐命令等を含む明示的な手続きを記述する必要はない。

10 15 【0079】また、ワークフローのルールに記述されている手順にしたがって実行してあるタスクにおいて、ユーザがそのタスクを終了していない中途の状態であっても、必要に応じてそれ以降のタスクを起動することができる。そして、中途のタスクにおいて、作成されるべき各成果が完成するごとに、すでに起動されている後続のタスクへその成果を自動的に渡すことがルールを変更することなくできる。これによって同じワークフローのルールの中で、タスクの実行順序に柔軟性を持たせることができる。

20 25 30 【0080】さらに、システムが解決できない例外がいずれのタスクで発生した場合においても、その例外に対する人間の対処による結果をシステムに取り込むことができ、その結果に基づいて、以降のワークフローを継続して実行できる。これにより、人間の対処をも含めたワークフロー全体の柔軟な管理および支援を行なうことが可能となるという効果がある。

【図面の簡単な説明】

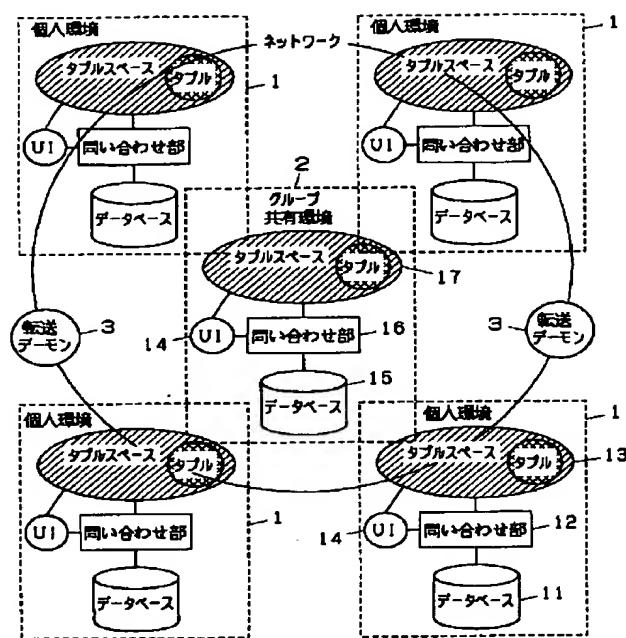
【図1】 本発明によるワークフロー支援システムの一実施例を示す構成図である。

35 【図2】 本発明のワークフロー支援システムにおける個人環境1の問い合わせ部12の動作の一例を示すフローチャートである。

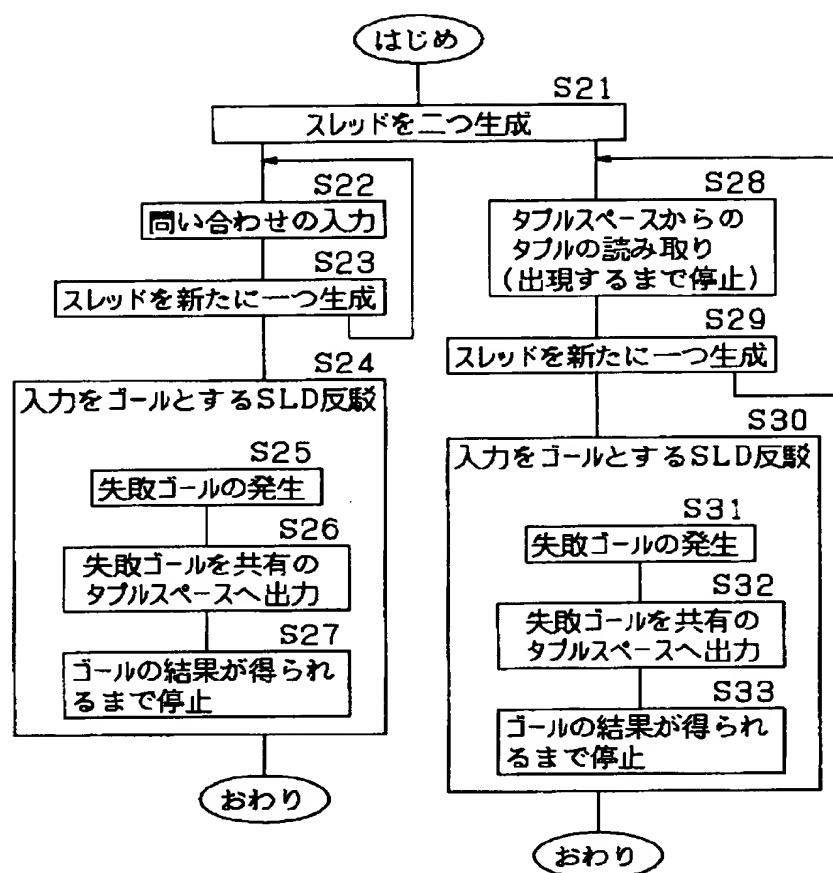
【符号の説明】

1…個人環境、2…共有環境、3…転送デーモンプロセス、11、15…データベース、12、16…問い合わせ部、13、17…タブルスペース、14…ユーザインターフェース。

【図1】



【図2】



フロントページの続き

(72) 発明者 申 吉浩

神奈川県横浜市保土ヶ谷区神戸町134番地
 横浜ビジネスパークイーストタワー 富士ゼロックス株式会社内